Response to Refutation of Aslam's Proof that $\mathbf{NP} = \mathbf{P}$

Javaid Aslam*
jaslam@softgalactic.com

June 28, 2009

Abstract

We present a resolution to the refutation provided by Ferraro et. al (arxiv.org, May 2009), for the proof of $NP = P$ in [Aslam, arxiv.org, March 2009]. We also provide a correct solution to the counter example and additional results that explain why some issues raised in the cited refutation are not quite valid.

1 Preserving ER over a VMP Set

The authors' conclusion is valid in pointing out the problem in maintaining the Edge Requirements (ER) of the VMPs over the VMP set, VMPSet, and which arises due to the inability of the data structure for storing a $VMPSet(a, b)$ between the two qualified mdags, called nconns, induce by the nodes a and b. However it must be noted that the authors [Fra09] imply that the $VMPAdd(VMPSet1(a, b), VMPSet2(a, b))$ operation is performed over the VMPs between two nodes a and b. This is not correct. The two VMPs implicitly refer to a common pair of mdags induced by a and b. A resolution to this problem is as follows.

This problem of preserving ER is resolved by performing the $AddVMP()$ operation *only* over the set of CVMPs (as opposed to over the set of VMPs). Note that the CVMPs behave essentially like an R-path, and thus will contribute at the most one edge resulting from the $SE(p)$ (the defn. in [Fra09]) of any CVMP, p, in a multiplication of two CVMPs. And then a perfect matching can always be represented as a unique sequence of CVMPs. This revision requires some additional concepts.

Atomic CVMP
Definition 1.1. A CVMP, p in $\Gamma(n)$, is called an atomic CVMP if p cannot be expressed as a sequence of two or more CVMPs.

We will revise the definition of $VMPset$ as follows.
Let $CVMPSet(a_i, b_j)$ be a representation for a set of CVMPs between a common pair of mdags at the node pair (a_i, b_j) in $\Gamma(n)$, mirroring the data structure $VMPSet(a_i, b_j)$ defined in [Asl08].

Let $VMPList(x, y)$ be a collection of VMPs between a common pair (d_x, d_y) of mdags at the node pair (x, y) in $\Gamma(n)$.
For the uniformity of representation each VMP in $VMPList(x, y)$ will be represented as $VMPSet(x, y)$ even though there is exactly one VMP in $VMPSet(x, y)$. (Note that the pair (d_x, d_y) is implied by the context of REDGE and SEDGE matrices [Asl08])

Now we define a list of *shortest* VMPs between two common mdags as follows.
Definition 1.2. A VMP list, $VMPList(a, b)$, is called atomic if, for all $VMPList(x, y)$ containing smaller VMPs, $|VMPList(x, y)| = 1 \leq |VMPList(a, b)|$. for every $VMPSet(x, y)$ in $VMPList(x, y)$, covered by some VMP in $VMPList(a, b)$.

Let $VMPSeq(x, y)$ be a sequence of atomic $VMPList$s between a common pair (d_x, d_y) of qualified mdags (called *nconn*, defined in [Asl08]) at the node pair (x, y) in $\Gamma(n)$, such that each VMP in a VMPList can multiply the adjacent VMP in the next VMPList in the sequence. That is,

$$VMPSeq(x, y) \stackrel{\text{def}}{=} < VMPList(x, a_1), VMPList(a_1, a_2), \ \cdots, \ VMPList(a_r, y) >,$$

such that $\forall p_i \in VMPList(a_i, a_{i+1})$, $p_0 p_1 \cdots p_r$ is a VMP in $\Gamma(n)$, where $r < n-1$, $a_0 = x, a_{r+1} = y$.

Lemma 1.3. *The ER of each atomic CVMP, p in a $CVMPSet(a, b)$ can always be maintained to be the same over $CVMPSet(a, b)$ for any bipartite graph.*

The proof will follow from the following constructs and algorithm for a revised $AddVMP()$ operation.

From the above definition of atomic CVMP and the parallel between an atomic CVMP and an R-edge, one can verify the following result

Lemma 1.4. *Each perfect matching in $\Gamma(n)$ is a sequence of at the most $(O(n))$ atomic CVMPs.*

This Lemma essentially tells us that the ER of each atomic CVMP can vary over the set of atomic CVMPs which constitute a perfect matching, while the ER of each atomic CVMP in a $CVMPSet(a, b)$ can be preserved.

The JoinVMP() and AddVMP() operations in Algorithm 3 in [Asl08] are to be modified to follow the following rules:

$$f_1 : VMPList(a, b) \times CVMPSet(b, c) \rightarrow CVMPSet(a, c) \tag{1.1}$$
$$f_2 : VMPList(a, b) \times VMPList(b, c) \rightarrow VMPSeq(a, c) \tag{1.2}$$
$$f_3 : CVMPSet(a, b) \times VMPList(b, c) \rightarrow VMPSeq(a, c) \tag{1.3}$$
$$f_4 : CVMPSet(a, b) \times CVMPSet(b, c) \rightarrow CVMPSet(a, c) \tag{1.4}$$

The mapping f_1 in (1.1) covers essentially the scenario given in the counter example in [Fra09]). We will now provide a correct solution to the counter example and then present the algorithms for the revised operations. Finally we present the proof of Lemma 1.3 and the correctness of the revised algorithm.

2 The Counter Example Re-visited

First we note that the mdags in $VMPSet(a, b)$ in [Asl08] are implied by the context, and thus all the VMPs are between the two mdags induced by the node pair (a, b) where the R- and S-edges are defined by the context given by the REDGE and SEDGE matrices.

Let $CVMPSet(c_3, c_8)$ [Fig. 1(b)] be an atomic CVMP already found such that both the CVMPs in $CVMPSet(c_3, c_8)$ have the same ER.
To make the technique explicitly clear, we add one more node pair (x, x) in the bipartite graph BG', giving the node $(x9, 1x)$ in $\Gamma(10)$. Note that without this additional node there is no common mdag pair for (the old) $VMPSet(c_1, c_3)$, and the refutation pointed out in [Fra09] does not really hold.

Let $VMPSeq(x, c_3)$ be formed as defined above, containing exactly one atomic $VMPList(x, c_3)$ which has two VMPSets.

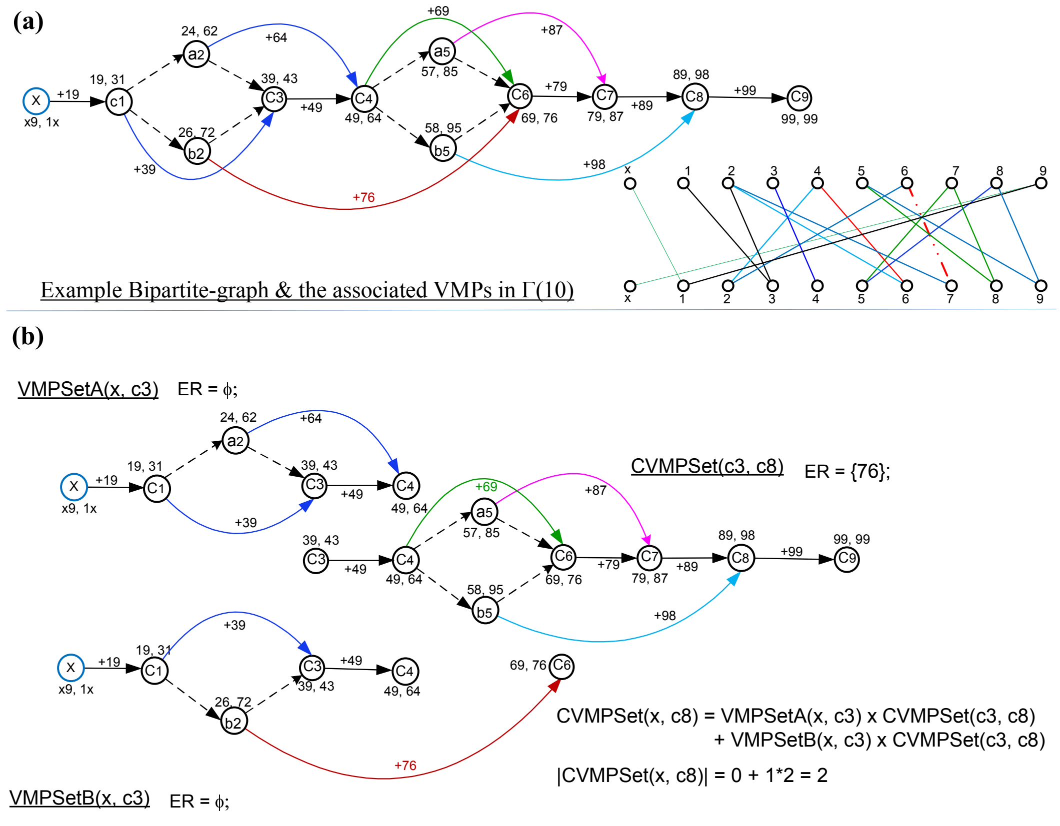

Figure 1: **Corrected Evaluation of VMPSets**

Then we perform the following two multiplications:

$$VMPSetA(x, c_3) \times CVMPSet(c_3, c_8), \tag{2.1}$$

$$VMPSetB(x, c_3) \times CVMPSet(c_3, c_8), \tag{2.2}$$

where the two VMPs, $VMPSetA(x, c_3)$ and $VMPSetB(x, c_3)$ are shown in [Fig. 1(b)].
To maintain the ER of each CVMP in the new CVMPSet, we add a newly formed product to the CVMPSet only if the ERs of all potentially affected nodes remain satisfied. (There is a further refinement to this logic covered in the following Algorithm 1) Since the first multiplication with $VMPSetA$ does not lead to satisfying the ERs of c_4 and c_6 both, we do not perform $AddVMP(VMPSetA(x, c_3) \times CVMPSet(c_3, c_8),\ CVMPSet(x, c_8))$. Therefore,

$$CVMPSet(x, c_8) = VMPSetB(x, c_3) \times CVMPSet(c_3, c_8),$$

and which gives $|CVMPSet(x, c_8)| = 2$.

Now we can formally define the function which determines when a VMP in a VMPList can be added to the new set of CVMPs.

Algorithm 1 ERQualifier ($vmpJumpEdgeList,\ allJumpEdgeList$)

Require: $vmpJumpEdgeList$ has all the R-edges specific to this VMP in $VMPList(a,b)$ and incident on any node in $CVMPSet(b,c)$.
Ensure: The ER of each new CVMP is independent of the R-edges not in $vmpJumpEdgeList$.

$affectedNodes \Leftarrow allJumpEdgeList - vmpJumpEdgeList$
$addVMP \Leftarrow$ **true**;
for all $(x,y) \in affectedNodes$ **do** {Evaluate the ER of each node}
  if $(SE(x,y) \in ER(y))$ **then**
    $addVMP \Leftarrow$ **false**;
    break;
  end if
end for
return $addVMP$;

The following algorithm builds a larger CVMP from a given pair of VMPList and CVMPSet, and maintains the ER of each new CVMP in the set to be the same.

Algorithm 2 buildCVMP ($VMPList(a,b),\ CVMPSet(b,c)$)

Require: Each CVMP in $CVMPSet(b,c)$ has the same ER
Ensure: Each CVMP in the new $CVMPSet(a,c)$ has the same ER

determine $allJumpEdgeList$ from $VMPList(a,b)$;
$newCVMPSet \Leftarrow \emptyset$;
for all $vmp \in VMPList(a,b)$ **do** {determine if a vmp can lead to the new $CVMPSet(a,c)$ }
  determine $vmpJumpEdgeList$ from vmp {specified by ERQualifier()}
  if (ERQualifier ($vmpJumpEdgeList,\ AllJumpEdgeList$)) **then**
    $tempCVMPSet \Leftarrow JoinVMP(vmp, CVMP(b,c))$
    $newCVMPSet \Leftarrow AddVMP(tempCVMPSet, newCVMPSet)$
  end if
end for
return $newCVMPSet$;

Let $P(m_a, m_b)$ be an atomic $VMPList(a,b)$ between a common pair of qualified mdags (called nconns), (m_a, m_b), at the node pair (a, b) in $\Gamma(n)$. Let $P(m_b, m_c)$ be a set of CVMPs between a common pair of mdags (m_b, m_c), at the node pair (b,c) in $\Gamma(n)$.

Property 2.1. *The number of VMPs in any atomic VMP list, $VMPList(a,b)$, is bounded by $O(n)$.*

Proof. Let $P(m_a, m_b)$ have r VMPs which are not R-paths, and consider a composition $P(m_a, m_b) \times P(m_b, m_c)$. The bound comes essentially from the upper bound on the number of R-edges that $P(m_b, m_c)$ can receive.

First we note that each VMP in $P(m_a, m_b)$ containing an S-edge gives rise to one jump edge that could span beyond the node b. Since R-paths can not contribute to any jump edges, there are at least $\Omega(r)$ jump edges which must be incident on $\Omega(r)$ nodes in the CVMP set $P(m_b, m_c)$, in order that each associated VMP multiplies each CVMP q in $P(m_b, m_c)$. Also, each such node in $P(m_b, m_c)$ must be covered by each $q \in P(m_b, m_c)$.

Second, we note that each node in any partition in $P(m_b, m_c)$ can receive at the most 2 R-edges. Therefore, $r \leq 2 * |q|$. Clearly, since $q \leq O(n)$, and the number of R-paths between two R-edges can not exceed $O(n)$, the result follows.

□

Correctness of Algorithm: buildCVMP()

First we prove Lemma 1.3.

Proof. (Lemma 1.3)
The proof is by induction on the size of $VMPList(a, b)$. Without loss of generality let there be exactly one R-edge, (x_i, y_i) from the ith VMP in $VMPList(a, b)$, where y_i is covered by each of the CVMPs in $CVMPset(a, c)$.

Basis: $|VMPList(a, b)| = 1$
This case is trivially true since the R-edge in *vmpJumpEdgeList* will multiply all the CVMPs in $CVMPset(b, c)$, and there is no other VMP in $VMPList(a, b)$ to affect the ER of the new CVMPs in $CVMPset(a, c)$.

$|VMPList(a, b)| = 2$
Note that each of the two R-edges can change the ER of the common CVMP which covers both, y_1 and y_2. Since exactly one of the VMPs in the list can chosen at a time, the only criteria for maintaining the ER of the new CVMPs would be to have $SE(x_1, y_1) \notin ER(y_1)$ and $SE(x_2, y_2) \notin ER(y_2)$. Or else, we will have at the most only one VMP from $VMPList(a, b)$.

Induction: $|VMPList(a, b)| = r + 1,\ r \geq 2$
Let $SE(x_i, y_i) \notin ER(y_i)$, $\forall i,\ 1 \leq i \leq r$. A new R-edge (x_{r+1}, y_{r+1}) from a new VMP in $VMPList(a, b)$ can maintain a common ER from all the new CVMPs only when additionally, $SE(x_{r+1}, y_{r+1}) \notin ER(y_{r+1})$. Otherwise, the new VMP gives rise to a new $CVMPSet(a, c)$ of size $|CVMPSet(b, c)|$.

□

Lemma 2.2. *Algorithm 2 correctly enumerates all the CVMPs in $CVMPset(a, c)$ between the two mdags induced by the node pair (a, c) in $\Gamma(n)$ for any bipartite graph in time $O(n^2)$.*

Proof. The correctness of enumeration depends on collecting all the "equally" satisfied ERs for each each CVMP in one set, and which follows from the correctness of $ERQualifier()$.

The time complexity $O(n^2)$ follows from Property 2.1 and $O(n)$ time complexity for each of the operations inside the FOR loop at line 3 in *buildCVMP*().

□

From the above proof one can easily see that each call to *buildCVMP*() can increase the size of the CVMPSet by a factor of $O(n)$ even in an incomplete bipartite graph. The procedure *buildCVMP*() produces a new set of CVMPs, $CVMPSet(a, c)$ of size either $|VMPList(a, b)| \times |CVMPSet(b, c)|$ or $|CVMPSet(b, c)|$ or zero.

3 Errors in Theorem 2 Proof in [Fra09]

This Theorem tries to point the basic results of Lemmas 5.8 and 5.9 of [Asl08]. The sufficiency of these results is essentially taken care of by the above revision, i.e., performing the AddVMP() operation only over a CVMP set as shown in the above algorithm buildCVMP().

The necessary conditions however still hold.
Note that when $[ER(x_i) \neq ER(x'_i)\ and\ e \in SE(A)] \Rightarrow$ No p in A can multiply all the VMPs in C, the ones covering x_i as well as those that cover x'_i. Multiplication is always tightly coupled with satisfying the edge requirement. And hence this composition is not valid for $A \times C$.

On the other hand, $ER(x_i) \neq ER(x'_i)$ would be the result incorrect inclusion of a VMP by an AddVMP() operation which produce C. Lemma 5.9 requires all the ERs of each node in any partition to be the same

essentially for a simultaneous ER satisfiability. Those VMPs that are not thus included in C are left to be satisfied by another multiplication composition.

The Lemma 3 in [Fra09] provides result on the exponentially many CVMPs having different SEs.
By Lemma 1.4 we need to maintain the ERs only over a set of atomic CVMPs, and each atomic CVMPSet can give rise to exactly one edge as SE, similar to an R-edge. Therefore, the number of CVMPs that are not atomic are irrelevant.

4 Acknowledgement

The author would like to thank the authors [Fra09] for their interest in this research, undertaking this paper as a formal course requirement and discovering the flaw in the algorithm. This was the first feedback of this level of details.

References

[Asl08] Javaid Aslam, *The Collapse of the Polynomial Hierarchy: NP = P*, `http://arxiv.org/pdf/0812.1385v9`, 2008.

[Fra09] Frank Ferraro, Garrett Hall and Andrew Wood, *Refutation of Aslam's Proof that NP = P*, `http://arxiv.org/pdf/0904.3912v2`, 2009.

